CpSc 421 Midterm 2 **November 5, 2008**

Do problem 0 and any three of problems 1-4.
If you attempt more than three of problems 1-4, please indicate which ones you want graded – otherwise, I'll make an arbitrary choice.

Graded on a scale of 100 points.
You can attempt from 105 to 110 points depending on which problems you choose. If you score over 100, you get to keep the extra credit.

0. (**5 points**) Your name: ________________________ Your student #: ______________

Question	Score
0	
TOTAL	

1. (**30 points**) Let $G = (V, \Sigma, R, \mathit{Expr})$ be the CFG with

$$\begin{array}{rcl} V & = & \{\mathit{Expr}, \mathit{Variable}, \mathit{Constant}, \mathit{Letter}, \mathit{Digit}\} \\ \Sigma & = & \{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}, \mathtt{a}, \mathtt{b}, \ldots, \mathtt{z}, \mathtt{+}, \mathtt{*}\} \end{array}$$

and rules

$$\begin{array}{rclclclcl} \mathit{Expr} & \to & \mathit{Variable} & \mid & \mathit{Constant} & \mid & \mathit{Expr}\ \mathtt{+}\ \mathit{Expr} & \mid & \mathit{Expr}\ \mathtt{*}\ \mathit{Expr} \\ \mathit{Variable} & \to & \mathit{Letter} & \mid & \mathit{Variable}\ \mathit{Letter} & \mid & \mathit{Variable}\ \mathit{Digit} & & \\ \mathit{Constant} & \to & \mathit{Digit} & \mid & \mathit{Constant}\ \mathit{Digit} & & & & \\ \mathit{Letter} & \to & \mathtt{a} & \mid & \mathtt{b} & \mid & \ldots & \mid & \mathtt{z} \\ \mathit{Digit} & \to & \mathtt{0} & \mid & \mathtt{1} & \mid & \ldots & \mid & \mathtt{9} \end{array}$$

(a) (**1**5 points) This grammar is ambiguous. Demonstrate this by drawing two different parse trees that generate the string

`x+12*y`

If you use the back side of one of these pages or one of the blank pages at the back, please write "*See page #*" where # is the page number here.

(b) (**1**5 points) Write an unambiguous grammar that generates the same language as G. You can just write the rules. If a variable in your grammar has the same rules as a variable in the grammar above, you can write

$$V \quad \to \quad \text{same as } G$$

2. (**35 points**) One of the two languages below is context-free (**20 points**), and the other is not (**15 points**). Identify which is which and justify your answers. For both languages, $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

$$
\begin{array}{rcl}
B_1 & = & \{s \mid (abs(\#a(s) - \#b(s)) \bmod 3) = 1\} \\
B_2 & = & \{s \mid \exists n.\ s = a^n b^{2n} a^n\}
\end{array}
$$

where $\#a(s)$ indicates the number of a's in s, $\#b(s)$ indicates the number of b's, and $abs(x)$ denotes the absolute value of x.

3. (**35 points**) Define

$$BalancedConcat(A, B) \quad = \quad \{s \mid \exists x \in A, y \in B.\ (|x| = |y|) \wedge s = xy\}$$

(a) (**15 points**) Show an example of a regular language for A and a regular language for B such that $BalancedConcat(A, B)$ is not regular. Give a short justification for your answer (my justification is one sentence long).

(b) (**20 points**) Show that for any regular languages A and B, $BalancedConcat(A, B)$ is context free. (my solution is eight sentences long).

4. (**35 points**) Let

$$B = \{M\#w\#n \mid M \text{ describes a Turing machine, } w \text{ describes a string, and } n \text{ is the binary representation of an integer, such that TM } M \text{ halts after at most } n \text{ steps when run with input } w.\}$$

(a) (**1**0 points)
Show that B is Turing decidable. You don't need a detailed proof. It is sufficient to sketch an algorithm for deciding whether or not a string is in B.
(My solution has four sentences.)

Because B is Turing decidable, there is a TM, M_B that decides B. Now, consider a non-deterministic TM, N_H, that on input $M\#w$ scans to the end of the input and appends $\#n$, where n is a string selected non-deterministically from $\{\mathtt{0}, \mathtt{1}\}^*$. Machine N_H then returns its head to the left end of the tape and runs machine M_B on the tape. If M_B accepts, then N_H accepts and if N_H rejects, then N_H rejects.

(b) (**10 points**) Draw the state transition diagram for a non-deterministic TM that appends $\#n$ to the end of its tape, where n can be the binary encoding of *any* integer. Your TM should start in state q_0 and transition to state p_0 when it has finished writing n.

(c) (**15 points**) Machine N_H accepts $M\#w$ iff machine M halts when run with input w. Thus, N_H recognizes language $HALT$. We can construct a deterministic Turing machine, M_H, that simulates N_H. We also know that $HALT$ is not decidable. Why isn't M_H (equivalently, N_H) a decider for $HALT$?